DOI: http://dx.doi.org/10.22441/fifo.2021.v13i1.009

Implementasi Algoritma Dijkstra dalam Pencarian Klinik Hewan Terdekat

Shandy[1], Canro Sigalingging[2], Jamal Jipesya[3], Yuwan Jumaryadi[4]

[1,2,3,4]Fakultas Ilmu Komputer, Program Studi Sistem Informasi, Universitas Mercu Buana
Jl. Meruya Selatan No. 1, Kembangan, Jakarta Barat 11650, Indonesia
[1]41816120052@student.mercubuana.ac.id, [2]41816120062@student.mercubuana.ac.id,
[3]41816120082@student.mercubuana.ac.id, [4]yuwan.jumaryadi@mercubuana.ac.id

Abstract

Technological developments that are increasing today can be felt in several aspects of life. Searching for the nearest veterinary clinic with the help of GPS technology will be easier than searching manually. This study aims to help animal owners who have difficulty in finding clinics and shops that sell various kinds of animal needs closest to the user's location. The development of this information system uses the Scrum model which consists of several stages, namely Product Backlog, Sprint Backlog, Sprint and Increment. The output of this developed information system can display the location and route of the nearest veterinary clinic based on Dijkstra's Algorithm. Based on the application trial research, the average time is 4 minutes 29 seconds in finding the nearest veterinary clinic. In addition, based on the SUS questionnaire given, a score of 76.96 was obtained, so the information system developed could assist users in finding the nearest veterinary clinic.
Keyword : Dijkstra's Algorithm, Scrum, Pet Owners, Animal Clinic Search.

Abstrak

Perkembangan teknologi yang meningkat sekarang ini dapat dirasakan pada beberapa aspek kehidupan. Pencarian klinik hewan terdekat yang dilakukan dengan bantuan teknologi GPS akan lebih mudah jika dibandingkan dengan pencarian secara manual. Penelitian ini bertujuan untuk membantu para pemilik hewan yang kesulitan dalam mencari klinik dan toko yang menjual berbagai macam kebutujan hewan terdekat dari lokasi pengguna berada. Pengembangan sistem informasi ini menggunakan model Scrum yang terdiri dari beberapa tahapan yaitu *Product Backlog*, *Sprint Backlog*, *Sprint* dan Increment. Hasil keluaran dari sistem informasi yang dikembangkan ini dapat menampilkan lokasi dan rute klinik hewan terdekat berdasarkan Algoritma Dijkstra. Berdasarkan penelitian uji coba aplikasi didapatkan waktu rata – rata 4 menit 29 detik dalam mencari klinik hewan terdekat. Selain itu, berdasarkan kuesioner SUS yang diberikan didapat nilai 76.96, sehingga sistem informasi yang dikembangkan dapat membantu *user* dalam pencarian klinik hewan terdekat.

Kata kunci : Algoritma Dijkstra, Scrum, Pemilik Pewan, Pencarian Klinik Hewan.

I. Pendahuluan

Penerapan teknologi yang semakin berkembang hampir dirasakan manfaatnya pada berbagai macam sendi kehidupan [1], yang salah satu diantaranya adalah dalam hal melakukan pencarian klinik hewan [2]. Ketika seseorang ingin mengunjungi klinik hewan langganan akan sangat sulit untuk memastikan klinik hewan tersebut apakah sedang memiliki jadwal yang sudah penuh atau tidak. Selain itu jika klinik hewan langganan penuh,

maka akan menjadi masalah tersendiri untuk mencari klinik hewan lalu memastikan langsung bahwa klinik hewan tersebut belum penuh.

Selain itu kebutuhan gizi hewan dan kebersihan hewan merupakan perhatian utama bagi para pemilik hewan peliharaan kesayangan [3]. Bagi pemilik hewan kesayangan, berbagai usaha akan dilakukan untuk menjaga hewan peliharaan nya senantiasa sehat seperti perawatan rutin ataupun pengobatan tertentu. Untuk memenuhi semua kebutuhan tersebut, maka dibuat aplikasi yang bertujuan dapat membantu para pemilik hewan mengatur jadwal kunjungan untuk melakukan perawatan atau pengobatan di klinik hewan langganan terdekat maupun bukan. Dimana klinik hewan dapat membuka kuota jumlah pengunjung dari aplikasi yang dapat datang di tiap periode jam nya sehingga tidak terjadi penumpukan atau antrian di dalam klinik.

Aplikasi ini juga harus memfasilitasi rute yang dapat ditempuh oleh para pengguna untuk menuju ke likasi klinik hewan dimanapun mereka berada. Hal ini diperlukan untuk mempermudah pemilik hewan peliharaan menuju klinik hewan yang belum pernah dikunjungi sebelumnya. Perkembagan teknologi saat ini telah mempengaruhi gaya hidup masyarakat. Segala kebutuhan bisa dilakukan secara *mobile* termasuk untuk perawatan hewan peliharaan kita. Kita bisa memesan jadwal perawatan hewan dan pengobatan hewan kesayangan tanpa harus datang ke klinik sebelumnya sehingga tentu saja hal ini memudahkan para pemilik hewan peliharaan ditengah kesibukannya.

1.1 Algoritma Dijkstra

Pada umumnya pencarian rute tercepat menggunakan Algoritma Dijkstra sebagai penyelesaian pencarian rute terdekat dari satu titik tertentu ke setiap titik lain di dalam graf. Operasi yang dijalankan pada Algoritma Dijkstra adalah dengan membuat satu pohon jalur Dijkstra terpendek yang diawali pada titik awal ”X”, sampai dengan titik tujuan ”Y” [4].

Algoritma Dijkstra banyak dipakai untuk berbagai aplikasi di dunia nyata yang membantu manusia dalam aktivitas keseharian seperti : layanan pemetaan digital di *Google Maps*, aplikasi jejaring sosial, jaringan telepon, menentukan rute *Internet Protocol* (IP) berdasarkan *Open Shortest Path First* (OSPF), *drone* yang dipakai untuk mengirimkan paket [5]. Dengan beragamnya penerapan Algoritma Dijkstra di berbagai bidang yang membutuhkan lintasan terpendek, membuat algoritma ini masih relevan untuk dipelajari dan dipakai hingga saat ini.

Perbedaan penelitian yang penulis lakukan ini dengan penelitian lain mengenai Algoritma Dijkstra adalah memanfaatkan Algoritma Dijkstra sebagai pencari klinik hewan terdekat dengan user dan menggunakan OS Android sebagai media platformnya. Hal ini bertujuan memanfaatkan Algoritma Dijkstra untuk penentuan lokasi klinik hewan terdekat dan menggunakan smartphone android sebagai alat menayangkan rute yang sudah diproses memakai Dijkstra. Selain itu kita juga dapat memaksimalkan fitur GIS yang dapat memberi info pada pemakai soal posisi dari klinik hewan dan *petshop*. Dalam sebuah perbandingan komparatif berdasarkan kompleksitas dan kinerja waktu aktual aplikasi antara yang menggunakan Algoritma Dijkstra dan algoritma Bellman – Ford didapatkan hasil bahwa Algoritma Dijkstra mengungguli algoritma Bellman – Ford [6]. Kombinasi antara Algoritma Dijkstra dengan rekayasa teknologi geolocation terbukti mampu menampilkan rute terpendek bagi para pengguna untuk menuju rumah ibadah, di mana dalam penelitian ini mengambil lokasi di kota Kudus [7]. Hasil implementasi Algoritma Dijkstra dalam eksekusi tercepat nya mampu menghasilkan keunggulan 0,0051 lebih cepat di bandingkan dengan algoritma *Ant Colony Optimization* dalam melakukan pencarian rute terpendek dan tercepat untuk angkutan umum TransJogja yang ada di kota Yogyakarta [8].

Dari penelitian mengenai Dijkstra di atas, implementasi Algoritma Dijkstra dapat digunakan sebagai penentu rute dari lokasi user ke klinik hewan terdekat supaya lebih presisi dan aktual sehingga proses yang dilakukan bisa jadi lebih cepat waktu.

1.2 Cara Kerja Algoritma Dijkstra

Algoritma Dijkstra beroperasi berdasarkan rute ke satu simpul optima pada setiap langkah. Jadi pada langkah ke – n, akan ada n node yang sudah diketahui jalur terpendek. Berikut ini merupakan prosedur Algoritma Dijkstra :

1. Pilihlah satu titik yang menjadi titik awal, kemudian beri nilai jarak kepada titik terdekat satu per satu.
2. Memberikan nilai 0 pada setiap titik awal dan nilai tak hingga terhadap titik lain.
3. Memberikan nama ”Node keberangkatan” di setiap titik awal dan ”belum terlewati” pada titik lainnya

4. Menghitung jarak selisih dari titik keberangkatan menuju titik terdekat atau tetangga nya. Pada jarak yang nilai nya lebih kecil di beri nama ”node terlewati”.
5. Simpan jarak terakhir yang nilai nya paling rendah.
6. Lalu berikan titik ”belum terlewati” dengan nilai jarak paling sedikit dari titik keberangkatan menjadi ”Node keberangkatan”
7. Jika semua titik sudah dilalui dan selesai maka nilai jarak terendah ialah jarak terakhir yang nilai nya paling kecil. Jika tidak, kembali ke langkah 4

Uji coba dikerjakan supaya memperoleh kesamaan jarak dari ”Node keberangkatan” sampai titik akhir pada pohon yang dibuat oleh Algoritma Dijkstra dengan jarak ukur. Guna memperoleh berapa persentase perbedaan bobot antar jarak yang akan diukur dengan jarak hasil ukur Dijkstra bisa dihitung dengan rumus :

$$S_j = \left[\frac{\sum_{i=1}^{N}(P_1 - P_2)}{N}\right] \times 100\%$$

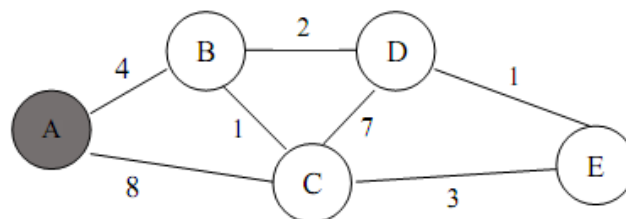
Keterangan :
Sj = Selisih jarak
P1 = Jarak yang akan diukur
P2 = Jarak yang sudah diukur Dijkstra

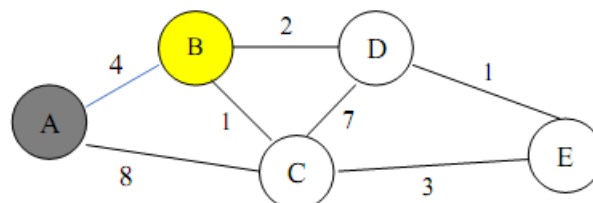
II. Metodologi Penelitian

2.1. Prosedur Penelitian

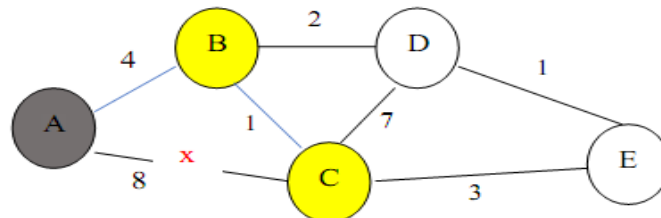
Penelitian ini menyatukan teknologi *mobile* dengan sistem operasi *android* untuk membantu kualitas pelayanan klinik hewan di mana saja. Peniliti merancang dan membuat aplikasi *mobile* dengan sistem operasi *android* yang mampu menampung semua informasi mengenai detail klinik hewan dan lokasi pengguna. Adapun untuk ketepatan penentuan rute antara user dan klinik hewan memakai Algoritma Dijkstra.

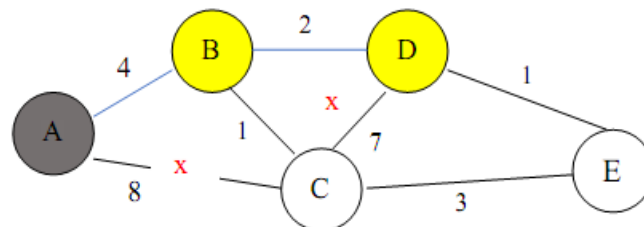
2.2. Proses Perhitungan Dijkstra

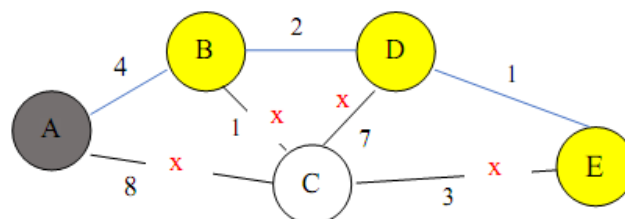
Dalam implementasinya Algoritma Dijkstra memerlukan ukuran dari titik berangkat dan titik akhir. Berikut ini merupakan *flow* dari Algoritma Dijkstra .

Gambar 1. Menentukan titik awal

Pada Gambar 1 diatas, node A adalah node awal dan node E merupakan node tujuan. Algoritma Dijkstra lalu mencari titik paling dekat yang tersambung dengan node A serta mempunyai nilai paling kecil.

Gambar 2. Bobot terendah terhubung dengan A

Pada Gambar 2 dapat diketahui bahwa nilai dari node A-C adalah 8, sedangkan nilai dari node A-B adalah 4, sehingga dapat diketahui bahwa node B memiliki nilai minimal. Setelah itu, Dijkstra akan mencocokkan nilai dari node yang tersambung dengan B melalui cara:
- Titik A-B-D = 6
- Titik A-B-C = 5

Dari nilai yang diperoleh ternyata nilai yang melalui titik A-B-C mempunyai nilai terkecil, jadi Algoritma Dijkstra melakukan kembali perhitungan nilai yang tersambung dengan C. Pada nilai D tidak dijadikan pilihan sebab nilai yang dipunyai lebih besar dari nilai A-B-D dan A-B-C seperti pada Gambar 2.

Gambar 3. Bobot terendah terhubung dengan C

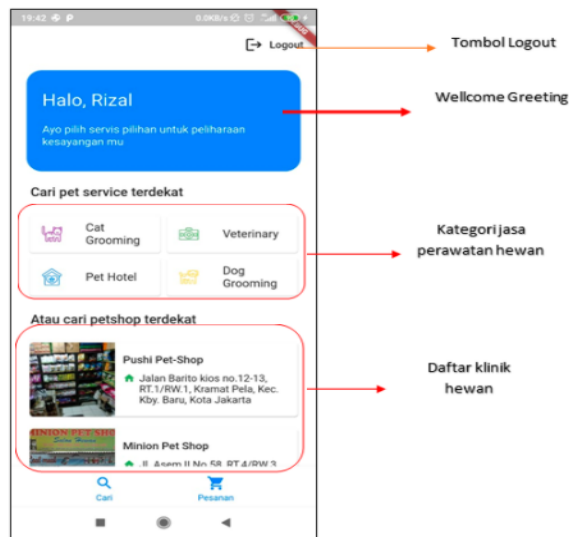
Gambar 3. menampilkan nilai yang melalui titik A-B-C-D adalah 12, sedangkan nilai yang melalui titik A-B-C-E adalah 8, selain itu nilai yang melalui titik A-B-C-E sudah tiba pada titik akhir, yaitu di titik E. Akan tapi A-B-C-E tidak bisa dipilih sebab masih ada rute lain yang mungkin mempunyai nilai lebih kecil yaitu A-B-D = 6 dan belum dibandingkan satu dengan yang lainnya oleh Dijkstra.

Gambar 4. Bobot terendah terhubung dengan D

Gambar 4. menampilkan titik D tersambung ke C juga E. Akan tetapi titik C tidak dipilih karena nilai yang melalui titik A-B-D-C melebihi nilai yang melalui titik A-B-C-E. Sehingga Algoritma Dijkstra langsung menghitung nilai yang melalui titik A-B-D-E.

Gambar 5. Hasil akhir jalur terpendek

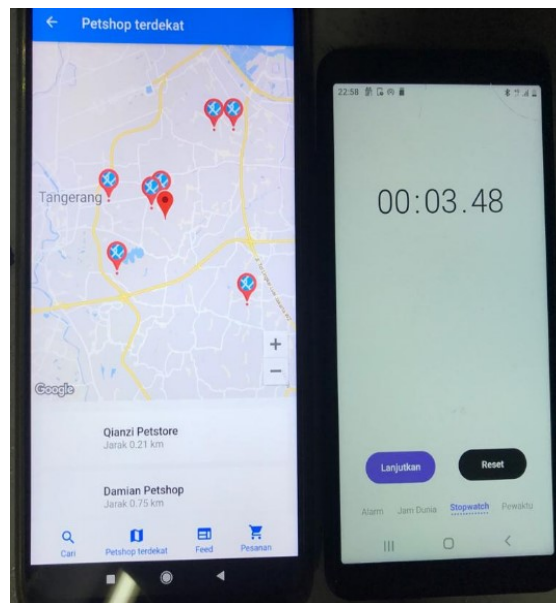
Pada Gambar 5 diperoleh nilai yang melalui titik A-B-D-E adalah 7, dan nilai yang melalui titik A-B-C-E adalah 8. Sehingga berdasarkan gambar tersebut dapat diketahui bahwa rute yang mempunyai nilai minimum adalah A-B-D-E, hal ini seperti digambarkan pada Gambar 5.

III. Hasil dan Pembahasan

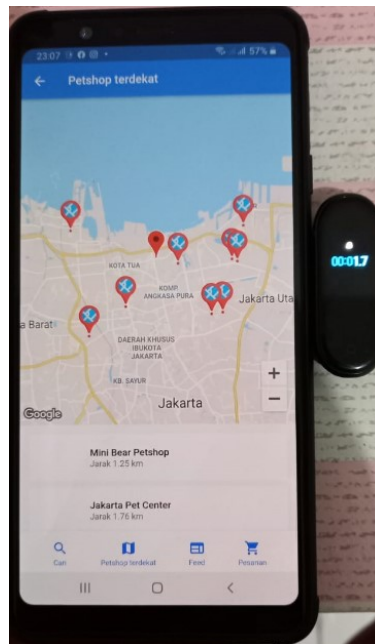
Berikut ini merupakan hasil dari perancangan yang telah dibuat pada tahap sebelumnya.

3.1. Halaman Utama

Gambar 6. Tampilan input data

Pada Gambar 6. menunjukan Halaman Utama dan daftar menu aplikasi seperti *Pet Hotel*, *Veterinary*, *Cat Grooming* dan *Dog Grooming*. Lalu ada klinik hewan yang menjadi favorit dan detail alamatnya.

3.2. Lama Waktu Pencarian di Cipondoh Indah

Gambar 7. Tampilan hasil pengukuran kecepatan

Pada Gambar 7. menunjukkan hasil pengukuran kecepatan pencarian klinik hewan terdekat dari lokasi pengguna di Kelurahan Cipondoh Indah.

3.3. Lama Waktu Pencarian di Pademangan

Gambar 8. Tampilan hasil pencarian rute

Pada Gambar 8. menunjukkan hasil pengukuran kecepatan pencarian klinik hewan terdekat dari lokasi pengguna di Kelurahan Pademangan.

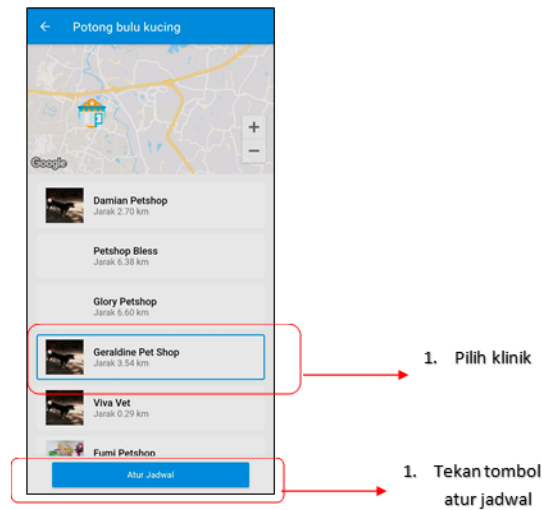
3.4. Lama Waktu Rata – rata Memuat

Tabel 1. Lama Waktu Rata – rata Memuat

Percobaan	Kel. Buaran Indah	Kel. Cipondoh Indah	Kel. Mangga Dua
I	00:09:16	00:02:38	00:04:03
II	00:04:13	00:03:48	00:02:09
III	00:08:25	00:04:39	00:01:07
Waktu rata-rata	**00:07:18**	**00:03:42**	**00:02:26**
		Total waktu rata - rata	**00:04:29**

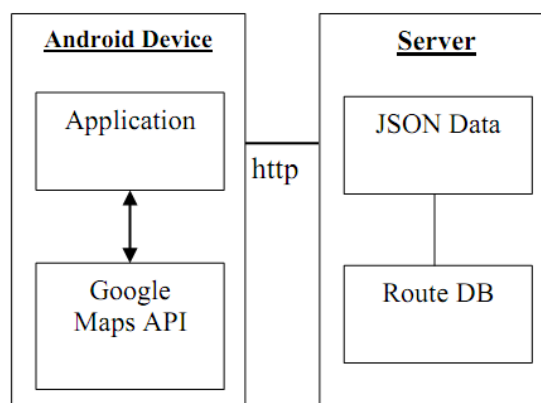
Pada Tabel 1. Menunjukkan rata - rata lamanya waktu yang dibutuhkan dalam menampilkan daftar klinik hewan terdekat dari titik pengguna berada. Pada table 1 diatas, dapat diketahui bahwa dilakukan 3 kali percobaan, dan didapatkan waktu rata-rata 4 menit 29 detik. Dari tabel 1 di atas dapat disimpulkan bahwa Algoritma Dijkstra berhasil menemukan jalur terpendek menuju klinik hewan dengan cepat.

3.5. Tampilan Hasil Rute

Pada Tabel 1. Menunjukkan rata - rata lamanya waktu yang dibutuhkan dalam menampilkan daftar klinik hewan terdekat dari titik

Gambar 9. Tampilan hasil pencarian rute

Pada Gambar 9 menunjukkan daftar klinik hewan yang dapat ditempuh sesuai dengan Algoritma Dijkstra sehingga dapat diperoleh lokasi klinik hewan terdekat. Setelah menentukan klinik hewan terdekat, maka user dapat melakukan registrasi untuk mengatur jadwal kunjungan ke klinik tersebut.

3.6. Proses Pengiriman Hasil ke Android

Aplikasi yang sudah diciptakan membuat permintaan untuk pencarian jarak memakai Algoritma Dijkstra

Gambar 10. Komunikasi antara aplikasi dan server

Pada Gambar 10 dapat dilihat mengenai komunikasi antara aplikasi dan server dalam pengembangan sistem informasi. Perhitungan Dijkstra dikerjakan di server lalu hasilnya dikirimkan lagi ke aplikasi dalam rupa jarak yang akan dilalui.

3.7. Pengujian Usabilitas

Peda penelitian ini dilakukan pengujian usabilitas untuk mengetahui kesesuaian sistem dengan kebutuhan pengguna (*user*) [9]. Pengujian usabilitas (*usability testing*) dilakukan dengan memberikan Kuesioner *System Usability Scale* (SUS) menggunakan Skala Likert. Pemberian Kuesioner ini dilakukan melalui Google Form. Hasil kuesioner yang didapatkan kemudian diolah dengan menggunakan standar Kuesioner *System Usability Scale* (SUS), dan nilai yang didapatkan adalah 76.96. Hal ini menunjukkan bahwa sistem yang dikembangkan

berada pada kategori "Good", dan nilai tersebut berada pada *grade* B [10]. Berdasarkan hasil kuesioner usabilitas dapat diketahui bahwa sistem informasi yang dikembangkan mendapat penilaian baik dari *user*.

Tabel 2. Kuesioner SUS

Pernyataan	Rata-rata
1. Saya merasa bahwa saya akan lebih sering menggunakan MyPetz	4.285714
2. Saya menilai bahwa seharusnya MyPetz tidak perlu serumit ini	2.357143
3. Saya menilai bahwa MyPetz mudah untuk digunakan	4.357143
4. Saya merasa bahwa saya akan memerlukan bantuan dari orang teknis agar dapat menggunakan MyPetz	2.071429
5. Saya menilai bahwa beberapa fungsi dari MyPetz telah terintegrasi dengan baik	4.214286
6. Saya merasa bahwa banyak ketidakkonsistenan dalam MyPetz	2.214286
7. Saya merasa bahwa MyPetz dapat dipelajari oleh sebagian besar orang dengan sangat cepat	4.714286
8. Saya menilai bahwa MyPetz sangat rumit untuk digunakan	1.857143
9. Saya merasa sangat percaya diri untuk menggunakan MyPetz	4.142857
10. Saya merasa perlu belajar banyak hal sebelum memulai menggunakan MyPetz	2.428571

IV. Kesimpulan

Dari penelitian dan percobaan yang sudah dikerjakan diperoleh kesimpulan bahwa Algoritma Dijkstra dapat membantu pengguna untuk menemukan klinik hewan terdekat dari lokasi pengguna berada saat itu, sehingga tujuan awal aplikasi ini dibuat dapat tercapai. Pengambilan koordinat lokasi menggunakan teknologi GPS pada Android dapat membantu *user* dalam memberikan arah petunjuk agar mencapai lokasi tujuan.

V. Daftar Pustaka

[1] D. Firdaus, B. Priambodo, and Y. Jumaryadi, "Implementation of push notification for business incubator," *Int. J. online Biomed. Eng.*, vol. 15, no. 14, pp. 42–53, 2019, doi: 10.3991/ijoe.v15i14.11357.

[2] C. M. Andrews *et al.*, "Mobile Technology in Veterinary Clinical Medicine," *J. Vet. Med. Res.*, vol. 2, no. 1, 2015.

[3] D. Pinardi, A. Gunarto, and S. Santoso, "Perencanaan Lanskap Kawasan Penerapan Inovasi Teknologi Peternakan Prumpung Berbasis Ramah Lingkungan," *J. Ilm. Peternak. Terpadu*, vol. 7, no. 2, p. 251, 2019, doi: 10.23960/jipt.v7i2.p251-262.

[4] H. Mehta, P. Kanani, and P. Lande, "Google Maps," *Int. J. Comput. Appl.*, vol. 178, no. 8, pp. 41–46, 2019.

[5] H. A. Musril, "Penerapan Open Shortest Path First (OSPF) Untuk Menentukan Jalur Terbaik Dalam Jaringan," *J. Elektro dan Telekomun. Terap.*, vol. 4, no. 1, p. 421, 2017, doi: 10.25124/jett.v4i1.989.

[6] S. W. G. Abusalim, R. Ibrahim, M. Zainuri Saringat, S. Jamel, and J. Abdul Wahab, "Comparative Analysis between Dijkstra and Bellman-Ford Algorithms in Shortest Path Optimization," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 917, no. 1, pp. 0–11, 2020, doi: 10.1088/1757-899X/917/1/012077.

[7] N. Azizah and D. Mahendra, "Geolocation dengan Metode Djikstra untuk Menentukan Jalur Terpendek Lokasi Peribadatan," *J. Sist. Inf. Bisnis*, vol. 7, no. 2, p. 96, 2017, doi: 10.21456/vol7iss2pp96-103.

[8] S. Sunardi, A. Yudhana, and A. A. Kadim, "Implementasi Algoritma Dijkstra dan Algoritma Semut Untuk Analisis Rute Transjogja Berbasis Android," *It J. Res. Dev.*, vol. 4, no. 1, pp. 1–9, 2019, doi: 10.25299/itjrd.2019.vol4(1).2483.

[9] N. Ani, H. Noprisson, and N. M. Ali, "Measuring usability and purchase intention for online travel booking: A case study," *Int. Rev. Appl. Sci. Eng.*, vol. 10, no. 2, pp. 165–171, 2019, doi: 10.1556/1848.2019.0020.

[10] A. Bangor, T. Staff, P. Kortum, J. Miller, and T. Staff, "Determining what individual SUS scores mean: adding an adjective rating scale," *J. usability Stud.*, vol. 4, no. 3, pp. 114–123, 2009.